

MEMORY HUB AND METHOD
FOR PROVIDING MEMORY SEQUENCING HINTS

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a
5 computer system having a memory hub coupling several memory devices to a processor
or other memory access device.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access
memory (“DRAM”) devices, to store data that are accessed by a processor. These
10 memory devices are normally used as system memory in a computer system. In a
typical computer system, the processor communicates with the system memory through
a processor bus and a memory controller. The processor issues a memory request,
which includes a memory command, such as a read command, and an address
designating the location from which data or instructions are to be read. The memory
15 controller uses the command and address to generate appropriate command signals as
well as row and column addresses, which are applied to the system memory. In
response to the commands and addresses, data are transferred between the system
memory and the processor. The memory controller is often part of a system controller,
which also includes bus bridge circuitry for coupling the processor bus to an expansion
20 bus, such as a PCI bus.

Although the operating speed of memory devices has continuously
increased, this increase in operating speed has not kept pace with increases in the
operating speed of processors. Even slower has been the increase in operating speed of
memory controllers coupling processors to memory devices. The relatively slow speed
25 of memory controllers and memory devices limits the data bandwidth between the
processor and the memory devices.

In addition to the limited bandwidth between processors and memory
devices, the performance of computer systems is also limited by latency problems that

increase the time required to read data from system memory devices. More specifically, when a memory device read command is coupled to a system memory device, such as a synchronous DRAM (“SDRAM”) device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices 5 can synchronously output burst data at a high data rate; the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory 10 hub architecture, a system controller or memory controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while 15 another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor.

Although computer systems using memory hubs may provide superior performance, they nevertheless often fail to operate at optimum speed for several 20 reasons. For example, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from latency problems of the type described above. More specifically, although the processor may communicate with one memory device while another memory device is preparing to transfer data, it is sometimes necessary to receive data from one memory device before the data from another memory 25 device can be used. In the event data must be received from one memory device before data received from another memory device can be used, the latency problem continues to slow the operating speed of such computer systems.

One technique that has been used to reduce latency in memory devices is to prefetch data, *i.e.*, read data from system memory before a program being executed 30 requests the data. Generally the data that are to be prefetched are selected based on a

pattern of previously fetched data. The pattern may be as simple as a sequence of addresses from which data are fetched so that data can be fetched from subsequent addresses in the sequence before the data are needed by the program being executed. The pattern, which is known as a “stride,” may, of course, be more complex.

5 Further, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from throughput problems. For example, before data can be read from a particular row of memory cells, digit lines in the array are typically precharged by equilibrating the digit lines in the array. The particular row is then opened by coupling the memory cells in the row to a digit line in respective
10 columns. A respective sense amplifier coupled between the digit lines in each column then responds to a change in voltage corresponding to the data stored in respective memory cell. Once the row has been opened, data can be coupled from each column of the open row by coupling the digit lines to a data read path. Opening a row, also referred to as a page, therefore consumes a finite amount of time and places a limit on
15 the memory throughput.

Finally, the optimal decision of whether or not to prefetch data (and which data to prefetch), as well as whether or not to precharge or open a row, and whether or not to cache accessed data, may change over time and vary as a function of an application being executed by a processor that is coupled to the memory hub.

20 There is therefore a need for a computer architecture that provides the advantages of a memory hub architecture and also minimizes the latency and/or throughput problems common in such systems, thereby providing memory devices with high bandwidth, high throughput, and low latency. Such a system would also desirably allow the operation of the memory hub to change over time.

25 SUMMARY OF THE INVENTION

According to one aspect of the invention, a memory module and method is provided including a plurality of memory devices and a memory hub. The memory hub contains a link interface, such as an optical input/output port, that receives memory requests for access to memory cells in at least one of the memory devices. The memory

hub further contains a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests. The memory hub further

5 is coupled to a system controller, the system controller operable to generate a memory hint. The memory hub further contains a memory sequencer coupled to the link interface and the memory device interface. The memory sequencer is operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface. The memory sequencer is further operable to

10 dynamically adjust operability responsive to the memory hint.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system according to one example of the invention in which a memory hub is included in each of a plurality of memory modules.

15 Figure 2 is a block diagram of a memory hub used in the computer system of Figure 1 according to an example of the invention.

Figure 3 is a schematic outline of a write command packet according to one example of the invention.

20 Figure 4 is a schematic outline of a read command packet according to one example of the invention.

Figure 5 is a block diagram of a memory hub used in the computer system of Figure 1 according to an example of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the invention is

25 shown in Figure 1. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is

typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory (“SRAM”). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a “North Bridge” or “memory controller.”

5 The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow
10 an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage
15 media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

 The system controller 110 is coupled to several memory modules 130a,b...n, which serve as system memory for the computer system 100. The memory modules 130 are preferably coupled to the system controller 110 through a high-speed
20 link 134, which may be an optical or electrical communication path or some other type of communications path. The high-speed link 134 may be either a bi-directional link, or it may include two separate bi-directional links, one of which couples signals from the system controller 110 to the memory modules 130 and the other of which couples signals from the memory modules 130 to the system controller 110. In the event the
25 high-speed link 134 is implemented as an optical communication path, the optical communication path may be in the form of one or more optical fibers, for example. In such case, the system controller 110 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication path. The memory modules 130 are shown coupled to the system
30 controller 110 in a point-to-point coupling arrangement in which a separate segment of

the high-speed link 134 is used to couple each of the memory modules 130 to either each other or to the system controller 110. However, it will be understood that other topologies may also be used, such as a multi-drop arrangement in which the single high-speed link (not shown) is coupled to all of the memory modules 130. A switching 5 topology may also be used in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

The high-speed link 134 serves as the path for communicating command, address and data signals between the system controller 110 and the memory modules.

10 The command, address and data signals can assume a variety of formats. However, in the embodiment shown in Figure 1, the command, address and write data signals are all embedded in memory packets that are transmitted from the system controller 100 to the memory modules 130. Memory packets containing read data signals are embedded in packets coupled from the memory modules 130 to the system controller 110.

15 Each of the memory modules 130 includes a memory hub 140 for controlling access to 32 memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory (“SDRAM”) devices. However, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may, of course, also be used. In the example 20 illustrated in Figure 1, the memory hubs 140 communicate over 4 independent memory channels 149 over the high-speed link 134. In this example, although not shown in Figure 1, 4 memory hub controllers 128 are provided, each to receive data from one memory channel 149. A fewer or greater number of memory channels 149 may be used, however. The memory hub 140 is coupled to each of the system memory devices 148 25 through a bus system 150, which normally includes a control bus, an address bus and a data bus.

A memory hub 200 according to an embodiment of the present invention is shown in Figure 2. The memory hub 200 can be substituted for the memory hub 140 of Figure 1. The memory hub 200 is shown in Figure 2 as being coupled to four 30 memory devices 240a-d, which, in the present example are conventional SDRAM

devices. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the 5 memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

Further included in the memory hub 200 are link interfaces 210a-d and 212a-d for coupling the memory module on which the memory hub 200 is located to a 10 first high speed data link 220 and a second high speed data link 222, respectively. As previously discussed with respect to Figure 1, the high speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the 15 high speed data links 220, 222. As well known, such circuitry includes transmitter and receiver logic known in the art. It will be appreciated that those ordinarily skilled in the art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present 20 invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

The link interfaces 210a-d, 212a-d are coupled to a switch 260 through a 25 plurality of bus and signal lines, represented by busses 214. The busses 214 are conventional, and include a write data bus and a read data bus, although a single bi-directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily skilled in the art that the busses 214 are provided by way of example, and that the

busses 214 may include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 200 to be connected in the system memory in a variety of configurations.

5 For example, the point-to-point arrangement, as shown in Figure 1, can be implemented by coupling each memory module 130 to either another memory module 130 or to the memory hub controller 128 through either the link interfaces 210a-d or 212a-d. This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons, including relatively low capacitance, relatively

10 few line discontinuities to reflect signals and relatively short signal paths. Alternatively, a multi-drop or daisy chain configuration can be implemented by coupling the memory modules in series. For example, the link interfaces 210a-d can be used to couple a first memory module and the link interfaces 212a-d can be used to couple a second memory module. The memory module coupled to a processor, or system controller, will be

15 coupled thereto through one set of the link interfaces and further coupled to another memory module through the other set of link interfaces. In one embodiment of the present invention, the memory hub 200 of a memory module is coupled to the processor in a multi-drop arrangement.

The switch 260 is further coupled to four memory interfaces 270a-d

20 which are, in turn, coupled to the system memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each system memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines,

25 represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the system memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the system memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with system memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The switch 260 may also includes arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the system memory device 240a-d to which it is coupled and receiving data signals from the system memory device 240a-d to which it is coupled. The write buffer 282 and the cache memory unit 284 include the normal components of a buffer and cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The memory devices used in the write buffer 282 and the cache

memory unit 284 may be either DRAM devices, static random access memory (“SRAM”) devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded or stand-alone devices.

5 The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read request. Using this approach, memory requests can be serviced out of order since an
10 earlier write request can be stored in the write buffer 282 while a subsequent read request is being serviced. The ability to buffer write requests to allow a read request to be serviced can greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order. For example, a series of write requests interspersed with read requests can be stored in the write buffer 282 to allow
15 the read requests to be serviced in a pipelined manner followed by servicing the stored write requests in a pipelined manner. As a result, lengthy settling times between coupling write request to the memory devices 270a-d and subsequently coupling read request to the memory devices 270a-d for alternating write and read requests can be avoided.

20 The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of
25 the system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

Further included in the memory hub 200 is a built in self-test (BIST) and diagnostic engine 290 coupled to the switch 260 through a diagnostic bus 292. The diagnostic engine 290 is further coupled to a maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the diagnostic engine 290 in order to perform memory channel and link diagnostics. For example, the user can couple a separate PC host via the maintenance bus 296 to conduct diagnostic testing or monitor memory system operation. By using the maintenance bus 296 to access diagnostic test results, issues related to the use of test probes, as previously discussed, can be avoided. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the diagnostic engine 290 should accommodate the standards of the maintenance bus 296, where such a standard maintenance bus is employed. For example, the diagnostic engine should have a maintenance bus interface compliant with the JTAG bus standard where such a maintenance bus is used.

Further included in the memory hub 200 is a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. The DMA engine 286 can implement conventional DMA operations well known by those ordinarily skilled in the art. The DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the

system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

The diagnostic engine 290 and the DMA engine 286 are preferably embedded circuits in the memory hub 200. However, including separate a diagnostic 5 engine and a separate DMA device coupled to the memory hub 200 is also within the scope of the present invention.

As mentioned above, the command, address and data signals are preferably coupled between the memory hub controller 128 and the memory modules 130 in the form of memory packets. In accordance with one embodiment of the present 10 invention, a "hint," which are bits indicative of the expected future performance of the memory modules 130, is embedded in the memory packets and coupled to one or more of the memory hubs 140 in the memory modules 130. The hint, or hints, modifies the behavior of one or more memory hubs 140, as explained in greater detail below. In particular, the hint modifies the memory sequencing based on information known to or 15 estimated by the controller 128. For example, the controller 128 may have access to addressing information such as the memory requestor or address stride.

In one example of an addressing hint, the controller 128 communicates a command placing the hub 140 in page mode and identifying a number of pages to keep open. In another example, the controller 128 provides a hint related to prefetching – 20 such as 1, 2, or 4 cache lines that will follow. In another example, the controller 128 communicates a stride to the hub 140 – such as skip the next 1, 2, or 4 cache lines. In another example of a hint, the controller 128 may indicate whether or not to place a particular cache line in a hub cache. Of course, other hints may be used, or other specific information provided with the hints described.

25 Fig. 3 depicts a write command packet 300 including a hint 301. The write command packet 300 is generated by the controller 128 and communicated to the hub 140. The packet 300 includes the hint 301 and a command code 302. The write command packet 300 further includes write data 310, write address information 305, and may include other information such as a tag 311, a stride 312, a reservation 313, a 30 length 314, and error check information 315.

Fig. 4 depicts a read command packet 350 including a hint 351. The read command packet 350 is generated by the controller 128 and communicated to the hub 140. The packet 350 includes the hint 351 and a command code 352. The read command packet 350 further includes read address information 355, and may include 5 other information such as a tag 361, a stride 362, a reservation 363, a length 364, and error check information 365.

Read and write command packets, such as the packets 300 and 350, are sent to the hub 140. One implementation of the hub 140 for receiving the packets 300 and/or 350 is shown in Fig. 5. A read or write packet is received from a link in 400. A 10 request decoder 405 receives the packet and decodes the request and any hint or hints, which are provided to request queue 410. The request decoder 405 further decodes a read address, and provides the read address to a comparator 415. A write buffer queue 420 further receives packets from the link in 400 and provides a write address to the comparator 415. The comparator 415 compares the read and write addresses, and 15 notifies the request queue 410 of any write conflicts. If the request queue 410 identifies a read buffer hit, it accesses a prefetch buffer 425 to fulfill the request. Requests and hints are provided to a memory sequencer 430 connected to a memory interface 435. The memory sequencer 430 acts on any hint information, and sends request over the memory interface 435. Memory read data are coupled into the prefetch buffer 425 for 20 storage, if appropriate.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.